Sagnix-0.9
User Guide

September 8, 2015

Contents

Install **1**

Fast lane . . . 1

Slow lane . . . 2

Download & Extract . . . 2

Recompile Locally . . . 3

Install . . . 3

Usage **3**

Usage with a File Browser or Editor as Frontend . . . 3

Usage from the Command Line . . . 3

Account Configuration . . . 3

End-to-end Crypto . . . 4

Syncronize with the server . . . 4

Read your Mail . . . 4

Usage with the Java Graphical Frontend . . . 4

About this Program **5**

Development Status **5**

Enhancing Security **5**

End-to-End Encryption . . . 5

Diffie-Hellman end-to-end key generation with one persistent key pair per node. (Use this one.) . . . 5

Diffie-Hellman end-to-end key generation with two ephemeral key pairs per line. . . . 6

Personal private end-to-end key exchange . . . 6

End-to-Server Encryption . . . 6

Download Integrity . . . 6

Operating System Trust Boundaries 6

Support & Feedback **6**

Install

We explain how to install and use Sagnix on top of Linux, UNIX, BSD, OS X, or BusyBox.

You need an up-to-date operating system, e.g. Debian 8 (Jessie), Ubuntu 14.04 LTS (Trusty Tahr), Linux Mint 17.1 (Rebecca), OS X Yosemite + Xcode, or OpenBSD 5.6. This is because older systems will probably not include the required C++14 compiler. "UNIX-kernel + BusyBox + make + g++ + emacs" should be sufficient as long as g++ is Version 4.8 upwards.

Fast lane

The straightforward way to install sagnix is to type the commands

```
wget sagnix.eu/si -O si
sh si
su    <enter computer root password>
sh si install
exit
```

This has been tested with Linux. On OS X, you would rather

```
curl sagnix.eu/si > si
sh si
su admin  <enter computer admin password>
sudo sh si install  <enter computer admin password>
exit
```

Afterwards it suffices to execute the comand `sagnix-installer` to downlad and install future updated versions. `cd; sudo sagnix-installer deinstall` would remove both the installation and the download.

The graphical equivalent would be to download the sagnix-installer tool file, make it executable, and double-click it for execution in a terminal emulator. (OS X shows a misleading error message if you fail to chmod +x the file before executing it.) Or you open a teminal emulator and execute `sh Downloads/si.tool`.

We recommend that you do not switch to root from the X Window System because this effectively breaks the root/non-root separation. Use a text login instead. Also, do not paste browser content into terminal windows. In case you wish to read the script as a security measure, do not use the `cat`, `more`, and `less` commands, because return and backspace characters may cause misleading output.

If the above does not work for you, follow the steps below.

Slow lane

Download & Extract

Get a shell, either by pressing Ctl-Alt-F1 and logging in in Text Mode, or by starting the terminal application of your graphical desktop environment. In case you need to switch back to Graphic Mode, Ctl-Alt-F7 or something similar will do on many systems.

Download the most recent sagnix*.pax archive file, at best the one matching your operating system. The latter might enable you to skip the recompilation step below. We suppose that the file is in the folder $HOME/Downloads, as is the default on many computers. Type the commands below, set in typewriter font. A newline (return key) after every line is implied.

If security is a concern, authenticate your download:

```
cd; cd Downloads
sha256sum sagnix*.pax
```

If the result is not the same as what you received by other means, stop here. Beside the sagnix*.pax file, create a directory to expand the archive in:

```
mkdir sagnix
```

Expand the archive using tar:

```
cd sagnix
tar -xf ../sagnix*.pax
```

You should *not* do this as root. Git users may type

```
git clone --bare git://server2.sagnix.eu/sagnix.git
sha256sum sagnix.git
git clone sagnix.git
cd sagnix
```

The git repository's hash sum will differ from the pax archive's hash sum.

Recompile Locally

```
sh configure
sh make
```

Refer to the readme.txt file and the technical documentation as needed.

Install

The following commands will copy the program into place and set file permissions accordingly. As the copying needs privileges, you will have to use `login`, `su`, or `sudo` as needed, e.g.

`'su' <admin-account>` <enter your admin password>
`'sudo' sh install .` <enter your admin password>
`Ctrl-d` (leave the admin-account)
(The quotes prevent alias substitution. Keystrokes may be easily logged when the X Window System is active.)

Usage

We suppose that you managed to get a sagnix account, and that by now, you have two passwords that may serve as pre-shared secret for encrypted client-server communication.

Usage with a File Browser or Editor as Frontend

You may use Finder, Thunar, Emacs, or whatever file browser as a frontend to the sagnix command line interface. Before, you must set up your in- and outboxes and credentials with the `sagnix-accounts.sh` command. Symlinks below your home directory (in your $HOME/Desktop directory, if any) will point to your in- and outboxes.

To send a file, copy it into your outbox, prefix its name with the destination account number, and invoke the `sagnix-sync.sh` command. It will re-appear in the recipients inbox prefixed with your account number when he calls `sagnix-sync.sh`.

An icon image file for `sagnix-sync.sh` is available at `/usr/local/maildm/doc`.

Usage from the Command Line

Account Configuration

`mailmod +<telephone number> -t -k -p -n defaultkey`

asks for the (-p) passwords (concatenated to a single one) and initialises local account data structures, including the client-server (-k) keyfile and the identification (-t) token for this (+<telephone number>) account.

The '-n defaultkey' option sets the key for end-to-end encryption to a publicly known default key. This is handy for a quick start, but you should change the default as needed, possibly enforcing keys with a limited scope. Copy a keyfile into the sagnix-keys folder and use the '-n <your keyfile>' option to mailmod, or specify a keyfile on a per-message basis.

End-to-end Crypto

Encrypt your file for someone.

```
startapp /usr/local/maildm /exe/mailout +1 to 2 $HOME/<your file>
```

Replace '+1' with your account number an '2' with the recepients account number. The command will also decrypt any pending incoming files. The following command will decrypt only.

```
startapp /usr/local/maildm /exe/mailout +1
```

Syncronize with the server

```
startapp /usr/local/maildm /exe/maildm +1
```

Read your Mail

```
startapp /usr/local/maildm /exe/maildm +1
startapp /usr/local/maildm /exe/mailout +1
ls $HOME/inbox
```

Usage with the Java Graphical Frontend

At the time of writing, the java graphical interface is not functional yet. Once it is, typing

```
hej
```

at the command line starts the graphical interface and offers configuration options.

In case you use the GNOME Desktop Environment, you should add the `hej` command to your main menu (using the `alacarte` / MainMenu program) and place the inbox among your bookmarks by adding the line
`file:///usr/local/maildm/var/inbox`
to the file
`/home/you/.config/gtk-3.0/bookmarks`.

Refer to the technical documentation if you wish to configure in greater detail or run sagnix using the command line.

About this Program

The software encrypts files and establishes a connection to the sagnix email- and file transfer service.

You may consult the About file for further information.

Development Status

Have a look at the changelog for recent information.

Computers that run BusyBox or Raspbian are partially supported.

Enhancing Security

End-to-End Encryption

The default configuration includes pseudo end-to-end encryption. That is, it requires that you trust us with not reading your messages and with not forging messages.

In case you do not, you need true end-to-end encryption. You will have to share end-to-end keys with the persons you communicate with. This can be done either by meeting personally, or by using a Diffie-Hellman key generation mechanism for end-to-end keys.

Diffie-Hellman end-to-end key generation with one persistent key pair per node. (Use this one.)

To generate an end-to-end key, you may exchange two public keys via the pseudo-end-to-end encrypted channel. Afterwards, you can use true end-to-end encryption. By default, a public-private key pair is generated when an account (=node) is created. To make this approach work, both ends of the line (=pair of nodes) send their public keys via the pseudo-encrypted channel. Then they import the other party's public key, that is, they move it from their inbox to their key repository. The next message sent is encrypted with true end-to-end encryption.

Besides the two-message overhead, this technique requires that you trust us with not forging messages. It does *not* reqire that you trust us with not reading your messages. Attempts to forge the key exchange can be easily detected by making a phone call. As far as I know, there is no legal way to compel us to forge messages, and we have no incentive to do so, so you should trust us with that. Technically, if our servers get hacked, this may happen nonetheless.

The remaining risk is that if both the server gets hacked and one endpoint gets hacked (compromising a persistent private key), then a message that has already been deleted at the endpoint may be reconstructed from mirrored server traffic and the endpoint persistent private key.

Diffie-Hellman end-to-end key generation with two ephemeral key pairs per line.

To mitigate the attack described in the previous partagraph, two supplementary Diffie-Hellman key pairs may be generated for each pair of accounts. The resulting per-line secret key may be ephemeral to any degee.

Personal private end-to-end key exchange

You may share private end-to-end keys in real life. To reduce complexity, you may use the same key for all you friends / colleagues / whoever. This method also covers the reading/forging/hacking scenarios, but it shifts a lot of responsibility to the user.

End-to-Server Encryption

Parts of the key for end-to-server encryption become known to your bank and our bank, and possibly to other service providers. Though, it will be difficult for them to get the whole key. If you do not trust two third parities with not conspiring against you, ask us to share a key via classic mail.

Download Integrity

We offer hard-copied hashes. (Shipping via classic mail at a low fee.) Soft-copies can be found on sslsites.de and server2.sagnix.eu.

Operating System Trust Boundaries

Operating systems usually come with a large list of trusted public keys, implementing an opt-out policy with respect to pushing trust boundaries. This is not what you want to have on mission critical hardware.

Support & Feedback

You may get help from support@sagnix.eu.

If you encounter any issues (bugs, usability, security, portability, language correctness regarding English and C++, style, math, protocols, you name it), please report to code@sagnix.eu.